



高等学校计算机教材

C#

实用教程 (第2版)

◎ 梁敬东 钱晓军 朱毅华 时跃华 赵青松 编著
◎ 郑阿奇 主编



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

013364004

TP312C-43

214-2

高等学校计算机教材

C#实用教程

(第2版)

郑阿奇 主编

梁敬东 钱晓军 朱毅华 时跃华 赵青松 编著



电子工业出版社

Publishing House of Electronics Industry



北航

C1671793

TP312C-43

214-2

200385210

内 容 简 介

本书以 Visual Studio 2010 为平台,系统介绍了 C#语言基础知识及其 Windows 应用系统开发过程。全书包含实用教程、习题、实验和综合应用实习四部分,全面覆盖了 C#课程的理论基础和实践教学环节。实用教程首先重点介绍 C#语言基础,然后通过大量实例系统地介绍 C#进行 Windows 应用系统开发;习题突出基本编程语法和基本概念(附录 A 提供参考答案);实验主要锻炼编程和应用能力,读者先跟着做,然后自己练习;综合应用实习主要训练 Windows 应用程序开发能力。为了方便教学,本书配有教学课件,所有实例和综合应用实例源文件以及数据库文件,均可在华信教育资源网 <http://www.hxedu.com.cn> 免费下载。

本书可作为大学本科、高职高专有关专业 C#语言课程教材,也可供广大 C#语言开发用户学习参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

C#实用教程 / 郑阿奇主编;梁敬东等编著. —2 版. —北京:电子工业出版社,2013.7

高等学校计算机教材

ISBN 978-7-121-20907-9

I. ①C… II. ①郑… ②梁… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 150541 号

策划编辑:袁 玺

责任编辑:章海涛 文字编辑:袁 玺

印 刷:北京市李史山胶印厂

装 订:北京市李史山胶印厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编:100036

开 本:787×1092 1/16 印张:23.5 字数:600 千字

印 次:2013 年 7 月第 1 次印刷

定 价:49.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

C#是以 Microsoft.NET 为平台全新设计的现代编程语言。C#语言简单易用, 高效快捷, 功能强大, 深受用户欢迎。目前, 已经成为 Windows 应用系统开发和 Web 应用系统开发的首选工具。

2008 年, 我们结合 C#应用开发和高校教学的实践经验, 以 Visual Studio 2005 为操作平台, 编写了《C#实用教程》一书, 并先后重印 8 次, 受到读者的广泛欢迎。

《C#实用教程》(第 2 版) **以 Visual Studio 2010 为平台**, 系统介绍 C#程序设计语言, 并在此基础上介绍 Windows 应用系统开发。本书在第 1 版的基础上, 对编程平台及实验部分进行了较大的修改和完善, 不断提升了教材可读性和可用性。全书包含**实用教程、习题、实验和综合应用实习四部分, 附录 A 为习题参考答案。**

实用教程基本思路如下:

(1) 首先介绍 Visual Studio 2010 开发环境, 通过简单的实例介绍 C#学习和开发操作过程。其中, **“第一个控制台应用程序”**为后面上机练习 C#语言及其程序设计准备条件; **“第一个 Windows 窗体程序”**用于学习 Windows 应用系统开发操作过程; **“第一个 Web 应用程序”**使读者简单了解 C#作为脚本语言的使用方法。

(2) C#语言介绍包括 **C#语言基础**和 **C#面向对象编程**等内容。通过**“C#综合应用实例”**训练 C#综合程序设计能力。在此基础上介绍了 **C#高级特性、C#线程技术**。

(3) C#语言 Windows 应用程序开发包括思路、步骤、窗体、常用控件、对话框、菜单、文档等。然后介绍图形图像编辑、文件操作和数据库应用, 每一部分都有综合应用实例。

(4) 本书采用 Visual Studio 2010 中自带的 **SQL Server 2008 Express 作为数据库开发环境**。同时, 为了适应各种应用环境, 在华信教育资源网 <http://www.hxedu.com.cn> 上**提供 C#对数据库操作方法的电子文档**。

习题主要突出基本编程语法和基本概念, 附录 A 提供了习题参考答案。

实验主要锻炼编程和应用能力, 读者先跟着做, 然后自己练习。

综合应用实习突出使用 C#进行 Windows 应用程序开发的方法, 先引导读者完成系统的基本功能; 然后提出新的要求, 由读者自己增加、修改其他功能。

一般来说, 在较短的时间内, 通过实用教程学习、习题练习, 特别是上机和综合应用实习后, 读者基本能够应用 C#完成基本的应用编程。

本书**配有教学课件、所有实例和综合应用实例源文件以及数据库文件**, 方便教学和学生模仿练习。任课教师可在华信教育资源网 <http://www.hxedu.com.cn> 免费注册下载。

本书由南京师范大学郑阿奇主编和定稿, 由南京农业大学梁敬东、钱晓军、朱毅华、时跃华和赵青松编写。参加本书编写的还有顾韵华、王洪元、刘启芬、丁有和、曹弋、徐文胜、殷红先、张为民、姜乃松、彭作民、高茜、陈冬霞、钱晓军、朱毅华、时跃华、周何骏、赵青松、王一莉、徐斌、王志瑞、孙德荣、周怡明、刘博宇、周怡君、郑进、刘毅、陈杰、刘友春等。还有其他一些同志对本书的编写提供了许多帮助, 在此一并表示感谢!

由于作者水平有限, 不当之处在所难免, 恳请读者批评指正。

作者邮箱: easybooks@163.com

编 者

目 录

第1部分 实用教程

第1章 C#概述	1
1.1 C#语言简介	1
1.1.1 C#的优势	1
1.1.2 .NET 框架	2
1.2 Visual C# (2010) 开发环境	3
1.2.1 Visual Studio 2010 简介	3
1.2.2 Visual C#环境安装与设置	4
1.2.3 Visual Studio 2010 项目管理	6
1.2.4 Visual Studio 2010 IDE 界面元素	7
1.3 最简单的 C#程序	13
1.3.1 C#项目的创建与分类	13
1.3.2 第一个控制台应用程序	15
1.3.3 第一个 Windows 窗体程序	16
1.3.4 第一个 Web 应用程序	17
第2章 C#语言基础	20
2.1 数据类型	20
2.1.1 值类型	20
2.1.2 引用类型	21
2.1.3 两者关系	22
2.2 变量与常量	24
2.2.1 常量	24
2.2.2 变量	26
2.2.3 使用举例	26
2.3 运算符与表达式	27
2.3.1 算术运算符	28
2.3.2 关系运算符	28
2.3.3 逻辑运算符	29
2.3.4 位运算符	30
2.3.5 赋值运算符	32
2.3.6 条件运算符	33
2.3.7 运算符的优先级与结合性	33
2.3.8 表达式中的类型转换	34

2.4	C#语句的结构	35
2.4.1	三种基本的结构化流程	35
2.4.2	分支语句	35
2.4.3	循环语句	39
2.4.4	跳转语句	41
2.5	数组	45
2.5.1	一维数组	45
2.5.2	多维数组	47
2.5.3	不规则数组	49
2.5.4	数组的遍历	51
2.5.5	数组应用举例	52
2.6	类	54
2.6.1	类的声明	54
2.6.2	类的成员	55
2.6.3	构造函数	57
2.6.4	析构函数	59
2.7	方法	60
2.7.1	方法的声明	60
2.7.2	方法的参数	63
2.7.3	静态方法与实例方法	69
2.8	接口	70
2.8.1	接口的概念	70
2.8.2	接口的实现	72
第3章	C#面向对象编程	74
3.1	什么是面向对象编程	74
3.1.1	思想渊源与编程哲学	74
3.1.2	面向对象程序设计三原则	75
3.2	类的封装与继承	77
3.2.1	属性封装	77
3.2.2	类的继承	81
3.2.3	衍生类的构造函数	84
3.3	多态的实现	86
3.3.1	方法重载	86
3.3.2	运算符重载	88
3.3.3	虚方法与方法覆盖	89
3.3.4	抽象类与抽象方法	94
3.4	C#系统的类型转换	96
3.4.1	复合数据类型	96
3.4.2	数值转换	100
3.4.3	枚举转换	102
3.4.4	引用转换	104

3.4.5	使用 Convert 转换	105
3.5	编程常用算法	106
3.5.1	C#语言对排序查找的支持	106
3.5.2	最常用的 3 种排序算法	108
3.5.3	迭代与递归算法	112
3.6	异常	114
3.6.1	异常与异常类	114
3.6.2	异常处理	115
3.7	C#综合应用实例	120
第 4 章	Windows 应用程序开发基础	126
4.1	开发步骤演示	126
4.1.1	建立项目	126
4.1.2	界面设计	127
4.1.3	属性设计	127
4.1.4	代码设计	127
4.1.5	运行调试	127
4.2	窗体及其特性	128
4.2.1	窗体的外观样式	128
4.2.2	窗体可见性控制	129
4.2.3	窗体的定位	131
4.3	Windows 常用控件	133
4.3.1	认识控件大家族	133
4.3.2	标签控件	135
4.3.3	按钮与文本框	137
4.3.4	图片框	140
4.3.5	选择控件及分组	141
4.3.6	列表类控件	143
4.3.7	状态显示控件	146
4.4	对话框应用	148
4.4.1	消息框	149
4.4.2	模式对话框	150
4.4.3	通用对话框	150
4.4.4	应用举例	154
4.5	Windows 文档程序设计	155
4.5.1	菜单设计	156
4.5.2	单文档界面 (SDI)	158
4.5.3	多文档界面 (MDI)	160
4.5.4	文档的打印	165
第 5 章	C#高级特性	167
5.1	集合与索引器	167

5.1.1	自定义集合	167
5.1.2	集合类	169
5.1.3	索引器	170
5.2	委托与事件	172
5.2.1	初识委托	172
5.2.2	为什么要使用委托	174
5.2.3	多播委托	179
5.2.4	事件	181
5.3	预处理命令	183
5.3.1	符号定义与条件编译指令	184
5.3.2	警告错误指令	185
5.3.3	代码块标识指令	186
5.4	组件与程序集	186
5.4.1	组件	186
5.4.2	程序集	186
5.5	泛型	190
第6章	C#线程技术	193
6.1	引入线程的动机	193
6.1.1	进程的主线程	193
6.1.2	主线程的局限性	194
6.1.3	多线程编程思路	194
6.2	线程的创建及状态控制	195
6.2.1	Thread 类的使用	195
6.2.2	线程的创建、启动和终止	197
6.2.3	线程的挂起与恢复	200
6.2.4	线程的状态和优先级	203
6.3	线程同步和通信	205
6.3.1	lock 关键字	205
6.3.2	线程监视器	206
6.3.3	线程间的通信	211
6.3.4	子线程访问主线程的控件	213
6.4	线程的管理和维护	214
6.4.1	线程池	215
6.4.2	定时器	215
6.4.3	同步基元 Mutex	215
6.5	线程的应用	216
6.5.1	实时 GUI	217
6.5.2	并发任务	219
第7章	C#图形图像编程	222
7.1	图形设计基础	222
7.1.1	GDI+简介	222

7.1.2	绘图坐标系	223
7.1.3	屏幕像素	223
7.2	画图工具及其使用	223
7.2.1	笔	223
7.2.2	画刷	225
7.2.3	Graphics 类	227
7.3	绘制图形	229
7.3.1	线条定位与选型	229
7.3.2	画空心形状	231
7.3.3	图形的填充	236
7.4	字体和图像处理	238
7.4.1	定义字体	238
7.4.2	文本输出	239
7.4.3	绘制图像	239
7.4.4	图像刷新	240
7.5	综合应用实例	241
第8章	文件操作	247
8.1	.NET 的文件 I/O 模型	247
8.2	管理文件夹和目录	248
8.2.1	操作文件夹	248
8.2.2	处理路径字符串	249
8.2.3	读取驱动器信息	251
8.3	文件的基本操作	252
8.3.1	文件的种类	252
8.3.2	创建文件	253
8.3.3	读/写文件	256
8.4	综合应用实例	260
第9章	数据库应用	267
9.1	数据库基础	267
9.1.1	关系模型	267
9.1.2	SQL 语言	269
9.1.3	使用 Visual Studio 2010 平台创建 SQL Server 数据库	270
9.2	ADO.NET 原理	272
9.2.1	ADO.NET 概述	272
9.2.2	ADO.NET 对象模型	273
9.2.3	数据集与离线访问	274
9.3	创建和测试连接	276
9.3.1	Connection 连接字符串	276
9.3.2	使用 Connection 对象	277
9.3.3	连接的测试	278
9.4	在线操作数据库	279

9.4.1 SQL 命令的封装	279
9.4.2 信息的即时呈现	281
9.4.3 数据库在线访问示例	282
9.5 数据库的离线访问	284
9.5.1 数据适配	284
9.5.2 数据集机制	285
9.5.3 数据库离线访问示例	289

第2部分 习 题

第1章 C#概述	292
第2章 C#语言基础	292
第3章 C#面向对象编程	298
第4章 Windows 应用程序开发基础	302
第5章 C#高级特性	303
第6章 C#线程技术	304
第7章 C#图形图像编程	304
第8章 文件操作	305
第9章 数据库应用	306

第3部分 实 验

实验1 C#概述	308
实验2 C#语言基础	309
实验3 C#面向对象编程	316
实验4 Windows 应用程序开发基础	324
实验5 C#高级特性	329
实验6 C#线程技术	332
实验7 C#图形图像编程	334
实验8 文件操作	337
实验9 数据库应用	338

第4部分 C#综合应用实习——学生成绩管理系统

P.1 主界面及功能导航	341
P.2 学生信息查询	342
P.3 学生信息修改	345
P.4 学生成绩录入	350
P.5 自己动手扩展系统功能	355
附录A 习题参考答案	356
第1章 C#概述	356
第2章 C#语言基础	356
第3章 C#面向对象编程	358

第4章	Windows 应用程序开发基础	359
第5章	C#高级特性	361
第6章	C#线程技术	362
第7章	C#图形图像编程	362
第8章	文件操作	363
第9章	数据库应用	364

第1部分 实用教程

第1章

C#概述

C#是.NET平台为应用开发而全新设计的一种现代编程语言,随着微软公司.NET战略进入开发人员的视野后,C#很快成为Windows应用开发语言中的宠儿。本章将提供给读者C#快速入门的指导,帮助读者对C#有一个准确的定位,并能对自己的学习做好安排。

1.1 C#语言简介

1.1.1 C#的优势

作为编程语言,C#是简单的、完全面向对象的且是类型安全的语言。最重要的是,它是一种现代编程语言。在类、名字空间、方法重载和异常处理等方面,C#去掉了C++中的许多复杂性,借鉴和修改了Java的许多特性,使其更加易于使用,不易出错。

下面列举了C#在设计上较其他语言的优势。

1. 简单性

C#在设计上的首要目标就是简单性。没有指针是C#的一个显著特性。在默认情况下,用户使用一种可操控的代码进行工作时,一些不安全的操作,如直接地进行内存存取,将是不允许的。

在C#中,不再需要记住那些源于不同处理器结构的数据类型,C#在CLR层面统一了它们,使得基于.NET平台上的不同语言拥有相同的数据类型。我们可以将每种类型看作一个对象,不管它是初始数据类型还是完全的类。

整型和布尔型是完全不同的,这意味着if判别式的结果只能是布尔型,如果是别的类型,编译器则会报错。混淆比较和赋值运算的错误将不会再次发生。

2. 现代性

许多在传统语言中必须由用户自己来实现或者没有的特性,都成为C#语言中的基础部分。增加的金融类型对于企业级编程语言来说是很受欢迎的一个附加类型,用户可以使用一个新的decimal类型进行货币计算。

3. 高度面向对象

C#语言面向对象的程度比Java还要高,除了支持面向对象的所有概念:封装、继承和多态外,

还增加了委托、事件、集合、泛型等诸多新概念,并且整个C#的对象模型是建立在.NET 虚拟对象系统(VOS, Virtual Object System)之上的,是基础架构的一部分,而不仅仅是编程语言的特性。

C#中没有孤立的全局函数、变量或常数。每个对象都必须封装在一个类中,要么作为一个实例成员(通过类的实例对象访问),要么作为一个静态成员(通过类型来访问),这使用户的代码具有更好的可读性,同时减少了发生命名冲突的可能。

多重继承的优劣一直是面向对象领域争论最多的话题之一,因此C#只允许继承一个基类,如果需要多重继承,可以使用接口。

4. 类型安全性

C#实施了最严格的类型安全机制来保护它自身及其垃圾收集器。因此,程序员必须遵守关于变量的一些规定,比如不能使用未初始化的变量,否则编译器将会警告用户;对于对象的成员变量,编译器负责将它们置零;当数组实际上只有 $n-1$ 个元素时,不可能访问到它“额外”的元素 n ,这使重写未经分配的内存成为不可能。C#允许在应用级或语句级检查算术运算操作中的溢出,当溢出发生时会出现一个异常。此外,C#中传递的引用参数都是类型安全的。

5. 版本处理

动态链接库 DLL (Dynamic Link Library) 是一种被编译为二进制代码的函数库,在程序运行时才被调入内存执行,而不是在编译时链接到可执行程序内部,这样就在二进制级别实现了代码共享,而不必在每个应用程序中编译一个副本,如果 DLL 中的代码更新了,只需替换相应的 DLL 文件即可。然而,这也带来了 DLL 版本的问题,不同版本的 DLL 可能与调用程序不兼容,同一版本的 DLL 也不能同时为不同的调用程序服务,造成应用程序出现无法预料的错误,于是不得不更换文件名来保存同一 DLL 的多个版本——这种混乱的状态被称为“DLL 地狱”。不过令人欣慰的是,C#提供了一种版本处理机制,使得程序员可以确保当他开发的类库升级时,会与已有的应用保持二进制级别上的兼容。

1.1.2 .NET 框架

.NET 框架(.NET Framework)可以建立.NET 应用程序。使用.NET 开发的程序需要在.NET Framework 下才能运行。.NET Framework 体系结构主要包括下列五大部分:

- (1) 程序设计语言及公共语言规范(CLS);
- (2) 应用程序平台(ASP.NET 及 Windows 应用程序等);
- (3) ADO.NET 及类库;
- (4) 公共语言运行时(CLR);
- (5) 程序开发环境(Visual Studio)。

.NET 框架的体系结构如图 1.1 所示。构建在 Windows 操作系统之上的是公共语言运行时(CLR),其作用是负责执行程序,提供内存管理、线程管理、安全管理、异常处理、通用类型系统与生命周期监控等核心服务。在 CLR 之上的是基类库,提供许多类与接口。

.NET Framework 类库是以命名空间(Namespace)方式来组织的,命名空间与类库的关系就像文件系统中的目录与文件的关系一样,如用于处理文件的类属于 System.IO 命名空间。

在.NET Framework 基础上的应用程序主要包括 ASP.NET 应用程序和 Windows Forms 应用程序,其中 ASP.NET 应用程序又包含了“Web Forms”和“Web Service”,它们组成了全新的 Internet 应用程序;而 Windows Forms 则是传统的窗口应用程序。

在.NET Framework 之上,无论采用哪种语言编写的程序,都先被编译成中间语言 IL, IL

经过再次编译后才生成机器码,完成 IL 到机器码编译任务的是 JIT (Just In Time) 编译器。上述处理过程如图 1.2 所示。



图 1.1 .NET 框架结构

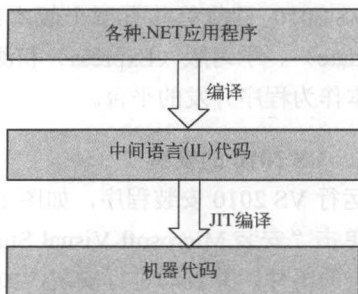


图 1.2 .NET 应用程序的编译过程

随着 .NET 技术的不断发展, .NET Framework 的发展也经历了几个阶段,从早期的 .NET Framework 1.0/1.1 发展到 2.0、3.0、3.5,功能越来越强。2010 年,随着 Microsoft 推出 Visual Studio 2010, .NET Framework 的版本又由 3.5 升级为 4.0。

.NET Framework 4.0 在 .NET Framework 3.5 的基础上进行了扩展,增加了很多新特性 (WCF, WPF, WF, LINQ 和 AJAX 等),使 .NET 技术更加强大和成熟。在利用 Visual Studio 2010 进行项目开发的时候,可以根据所要使用的特性选择所采用的 .NET Framework 版本,实现与早期版本的兼容。

1.2 Visual C# (2010) 开发环境

1.2.1 Visual Studio 2010 简介

Microsoft Visual Studio 是微软 .NET 平台的集成开发环境 (IDE),其功能强大、整合了多种开发语言 (包括 Visual Basic.NET、Visual C++.NET、Visual C#等),集代码编辑、调试、测试、打包、部署等功能于一体,大大提高了开发效率。

Visual Studio 2010 (简称 VS 2010) 是这个平台的最新版本,于 2010 年 4 月 12 日上市 (图 1.3 为该产品 LOGO),其 IDE 界面被重新设计和组织,变得更加简单明了。VS 2010 同时带来了 .NET Framework 4.0、CTP (Community Technology Preview),并且支持开发面向 Windows 7 的应用程序,数据库方面除了 Microsoft SQL Server 以外,它还支持 IBM DB2 和 Oracle 数据库。

新版的 Visual Studio 2010 实现了九大功能创新:

- (1) C# 4.0 中的动态类型和动态编程;
- (2) 多显示器支持;
- (3) 支持 TDD;
- (4) 支持 Office;
- (5) Quick Search 特性;
- (6) C++ 0x 新特性;
- (7) IDE 增强;
- (8) 使用 Visual C++ 2010 创建 Ribbon 界面;
- (9) 新增基于 .NET 平台的函数型程序设计语言 Visual F#。

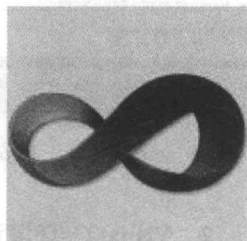


图 1.3 VS 2010 LOGO

1.2.2 Visual C#环境安装与设置

VS 2010 目前总共有五个版本: 专业版 (Professional)、高级版 (Premium)、旗舰版 (Ultimate)、学习版 (Express) 和测试版 (Test)。其中, 旗舰版的功能最为强大, 本书将以该版本作为程序开发的平台。

1. VS 2010 的安装

运行 VS 2010 安装程序, 如图 1.4 所示。

单击“安装 Microsoft Visual Studio 2010”, 进入安装向导 (图 1.5), 安装程序会自动加载所需的组件, 并自动进行安装, 如图 1.6 所示。这个过程持续时间较长, 请耐心等待!

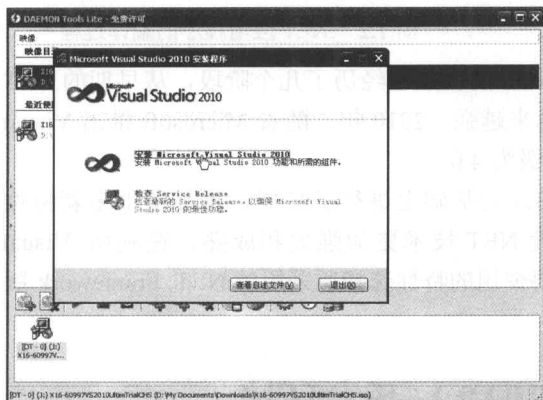


图 1.4 启动安装程序

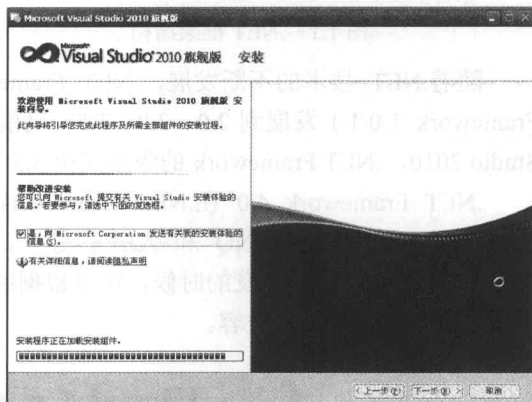


图 1.5 安装向导

安装过程结束后, 会出现“完成页”界面, 如图 1.7 所示, 显示安装成功。单击【完成】按钮, 结束安装。

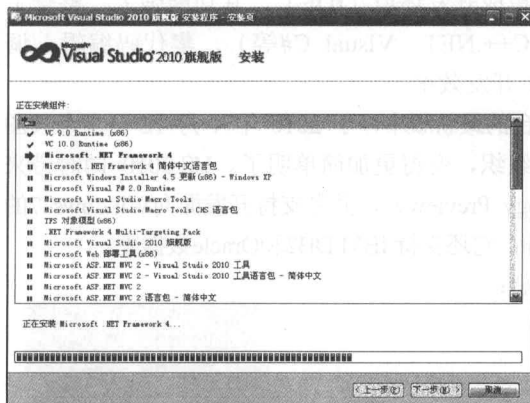


图 1.6 正在安装组件

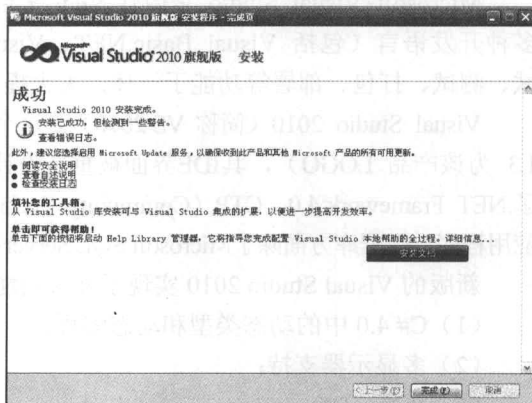


图 1.7 安装成功

2. Visual C#环境设置

安装完成后, 初次启动 VS 2010 程序, 会弹出【选择默认环境设置】对话框, 用户可在该界面自定义开发环境, 如图 1.8 所示, 这里我们选择“Visual C#开发设置”, 设置成 C#的编程环境。单击【启动 Visual Studio】按钮, 进入 Visual C#开发环境。

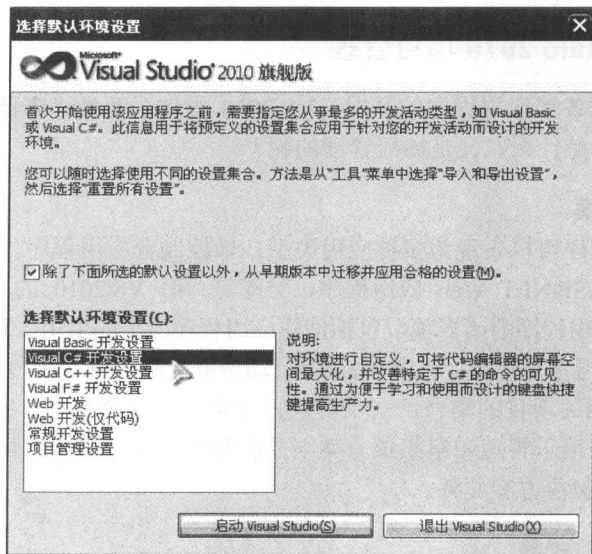


图 1.8 设置为 C# 开发环境

3. Visual C# 开发环境

经过配置后，打开 VS 2010 主窗口，显示“起始页”界面，如图 1.9 所示。

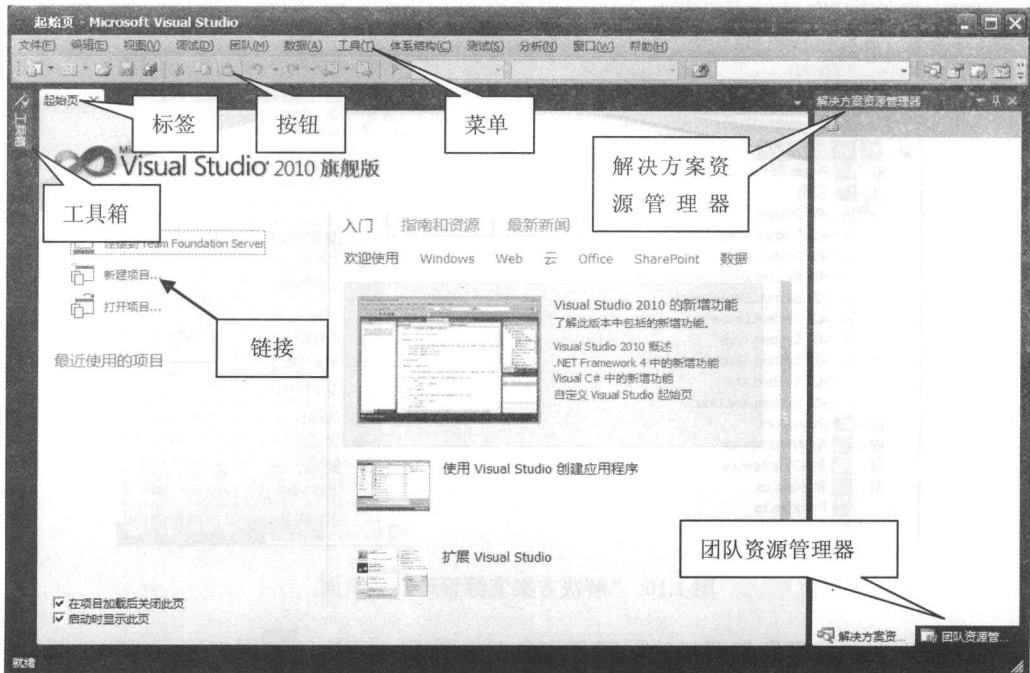


图 1.9 “起始页”界面

在“起始页”界面中，允许用户打开或新建项目。若要打开已有项目，可单击最近的项目列表中的某个项目名称；也可以依次单击菜单【文件】→【打开】→【项目/解决方案】，在弹出的【打开项目】对话框中选择要打开的项目。

1.2.3 Visual Studio 2010 项目管理

为了能有效地管理各类应用程序的开发, VS 2010 提供了两类“容器”: 一是**项目**, 二是**解决方案**。那么, 它们是什么? 又是如何管理的呢?

1. 项目与解决方案

VS 2010 开发的程序可以表现为多种应用类型, 如控制台应用程序、Windows 窗体应用程序、WPF 应用程序、ASP.NET Web 应用程序、类库等。而 VS 2010 的“项目”以逻辑方式管理、生成和调试构成应用程序的诸多项, 包括创建应用程序所需的引用、数据链接、文件夹和文件等。“项目”的输出通常是可执行程序(.exe)、动态链接库(.dll)文件或模块等。

解决方案是一类相关项目的集合, 一个解决方案可包含多个项目。VS 2010 还为解决方案提供了指定的文件夹, 用于管理和组织该方案下的各种项目和项目组。同时, 在该文件夹下还有一个扩展名为.sln 的解决方案文件。

2. 解决方案资源管理器

作为查看和管理解决方案、项目及其关联项的界面, “解决方案资源管理器”是 VS 2010 开发环境的一部分。它将方案中所有关联的项以“树视图”的形式分类显示。针对 Visual C#, 这些项包括 Properties (程序集属性)、引用(名字空间)、Resources (项目用到的资源, 如图片、音频、视频等)和.cs 文件(源文件)等, 单击节点名称图标前的“+”或“-”符号, 或双击图标, 将显示或隐藏节点下的相关内容, 如图 1.10 (左) 所示。

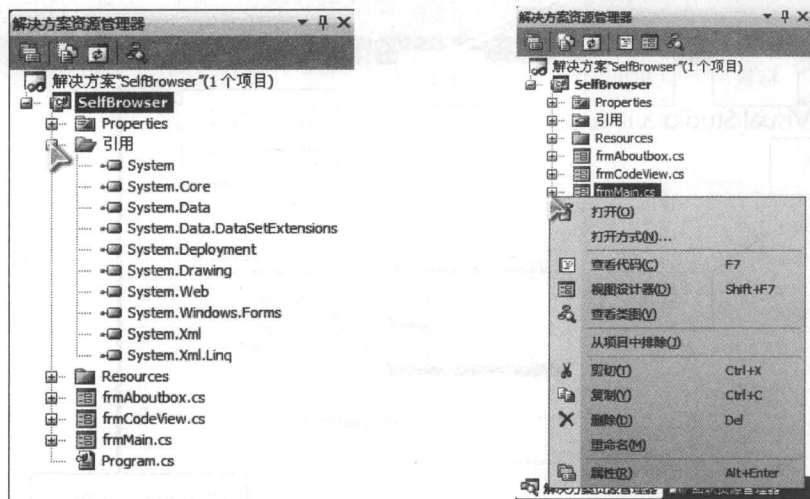


图 1.10 “解决方案资源管理器”页面

“解决方案资源管理器”窗口的顶部有几个工具图标。其中,  用来显示“树视图”中所选项的相应“属性页”对话框;  用来显示所有的项, 包括那些已经被排除的项和在正常情况下隐藏的项; 而  用来启动“类设计器”, 显示当前项目中类的关系图。

需要说明的是, 选择的节点项不同, 对应的窗口顶部出现的工具图标也不同。同时, 右键单击节点显示的快捷菜单也各不相同。例如, 右键单击 `frmMain.cs` 节点, 弹出如图 1.10 (右) 所示的快捷菜单, 从中可选择相应的命令和操作。